

(12) **EUROPÄISCHE PATENTANMELDUNG**

(43) Veröffentlichungstag:
20.06.2001 Patentblatt 2001/25

(51) Int Cl.7: **H03M 13/15, H04L 1/00**

(21) Anmeldenummer: 00127018.0

(22) Anmeldetag: 09.12.2000

(84) Benannte Vertragsstaaten:
**AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU
MC NL PT SE TR**
Benannte Erstreckungsstaaten:
AL LT LV MK RO SI

(72) Erfinder:
• **Witte, Franz-Otto, Dr. Ing.**
79312 Emmendingen (DE)
• **Zens, Volker, Dipl.-Ing.**
79211 Denzlingen (DE)

(30) Priorität: 17.12.1999 DE 19960923

(74) Vertreter: **Sauer, Wolfgang, Dipl.-Ing.**
MICRONAS GmbH,
Patentabteilung,
Hans-Bunte-Strasse 19
79108 Freiburg i.Br. (DE)

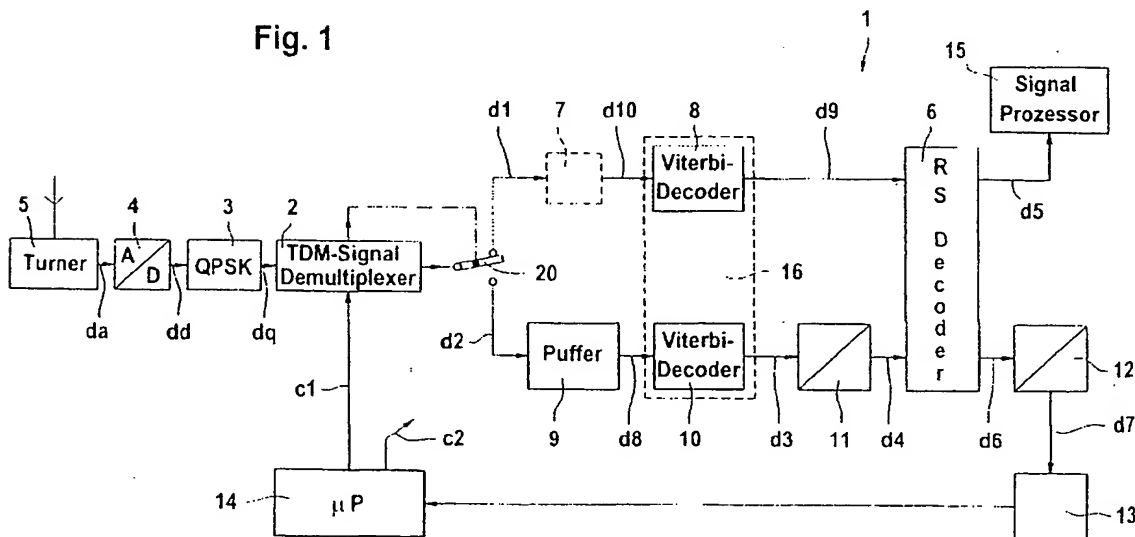
(71) Anmelder: **Micronas GmbH**
79108 Freiburg (DE)

(54) **Einrichtung zur Datenumsetzung für einen Reed-Solomon Dekodierer**

(57) Einrichtung (1) für einen Reed-Solomon Dekodierer (6), die mit einem ersten und zweiten Datenstrom (d1, d2) gespeist ist. Der erste Datenstrom (d1) ist dem Reed-Solomon Dekodierer (6), dessen Dekodierungsverfahren durch ein erstes primitives Polynom definiert ist, direkt oder über eine Zwischenverarbeitungsstufe

(7, 8) zugeführt. Der zweite Datenstrom (d2) ist einem Codeumsetzer (11) direkt oder über eine Zwischenverarbeitungsstufe (9, 10) zugeführt. Der Codeumsetzer (11) führt eine isomorphe Abbildung durch. Nach dem Reed-Solomon Dekodierer (6) findet eine zweite, zur ersten inverse isomorphe Abbildung dieser umgesetzten Daten mittels eines zweiten Codeumsetzers (12) statt.

Fig. 1



Beschreibung

[0001] Die Erfindung betrifft eine Einrichtung für einen Reed-Solomon Dekodierer, der im Zeitmultiplexbetrieb aus einer Datenquelle mit einem ersten und zweiten Datenstrom, deren Daten jeweils einem ersten und zweiten Reed-Solomon Code entsprechen, gespeist ist. Derartige Datenströme werden zunehmend bei der digitalen Übertragung von Signalen aus dem Audio- oder Videobereich verwendet. Neben der Codierung des eigentlichen Signalinhalts, der im zeitlichen Multiplexbetrieb mit anderen Signalen übertragen wird, sind auch Informationen zu übertragen, die zum Auffinden und Dekodieren des gewünschten Signals innerhalb des empfangenen Datenstroms dienen. Die im Zeitmultiplexbetrieb aneinandergefügte Signaldaten werden auf der Senderseite blockweise zur Datensicherung nach der ersten und die Steuerinformationen nach dem zweiten Reed-Solomon Code umgesetzt und bilden somit den ersten und zweiten Datenstrom. Die Übertragung dieser beiden Datenströme erfolgt seriell im Zeitmultiplexbetrieb. Das Format, in dem diese Datenströme übertragen werden, enthält in einem Kopfteil den zweiten Datenstrom mit den Kanal- und Steuerinformationen und in einem nachfolgenden Datenblock den ersten Datenstrom mit den Signalinhalten. Der erste Datenstrom wird in einigen Anwendungen als „Broadcast Channel“ (=BC) und der zweite Datenstrom als „Time Slot Control Channel“ (=TSCC) bezeichnet, in der folgenden Beschreibung wird diese spezielle Bezeichnung jedoch nicht weiter verwendet. Durch die vorgegebene Anordnung können die beiden Datenströme auf der Empfängerseite wieder leicht getrennt werden. Die nachfolgende Verarbeitung erfolgt mit unterschiedlicher Geschwindigkeit, je nachdem ob der erste oder zweite Datenstrom zu verarbeiten ist. Im zweiten Datenstrom sind alle aufeinanderfolgenden Daten wichtig und müssen daher alle erfaßt und verarbeitet werden. Im ersten Datenstrom, der zwar die gleiche Datenrate wie der zweite Datenstrom aufweist, müssen nicht alle aufeinanderfolgenden Daten oder Datenwörter erfaßt und verarbeitet werden, sondern nur diejenigen, die für den zu empfangenden Kanal relevant sind. Bei einer Vielzahl von Kanälen, die im zeitlichen Multiplexbetrieb übertragen werden, folgen auf ein relevantes Datenwort eine Vielzahl von nicht zu beachtenden Datenwörtern. Dadurch reduziert sich die erforderliche Verarbeitungsgeschwindigkeit im entsprechenden Verhältnis.

[0002] Unterschiedliche Codierungen nach dem Reed-Solomon-Verfahren unterscheiden sich in der Regel dadurch, daß die zugehörigen endlichen Körper, die auch als Galois-Körper (angelsächsisch: Galois-Field) bezeichnet werden, durch unterschiedliche primitive Polynome und die unterschiedlichen Parameter der codegenerierenden Polynome definiert sind. Die grundsätzliche Codierung und die Dekodierung derartiger Reed-Solomon Codes ist nicht Gegenstand der Erfindung. Beides ist bekannt und in einschlägigen Fachbüchern ausführlich beschrieben, vergleiche beispielsweise: Richard E. Blahut, „Theory and Practice of Error Control Codes“, 1983 bei Addison-Wesley Publishing Company, Inc., ca. 500 Seiten, ISBN 0-201-10102-5 oder Martin Bossert, „Kanalcodierung“, 1992 bei B. G. Teubner Stuttgart, ca. 283 Seiten, ISBN 3-519-06143-0. Eine weitere Sicherung gegenüber Übertragungsfehlern kann mit zusätzlichen Codierungsverfahren erreicht werden, beispielsweise einer vorwärtsgerichteten Fehlerkorrektur nach Viterbi.

[0003] Zur Verarbeitung von zwei Datenströmen, denen unterschiedliche Reed-Solomon Codes zugrunde gelegt sind, sind zwei voneinander unabhängige Reed-Solomon Dekodierer erforderlich. Und entsprechend ist für jeden weiteren Datenstrom ebenfalls ein weiterer Reed-Solomon Dekodierer erforderlich. Wegen der hohen Datenrate der zu verarbeitenden Daten im Echtzeitbetrieb sind wesentliche Teile der beiden Reed-Solomon Dekodierer als Hardware-Schaltung in einem speziell ausgebildeten GK-Prozessor („GK“ steht für Galois-Körper, im angelsächsischen Sprachbereich folgt daraus die Bezeichnung „Galois-Field Processor“ oder „GF-Processor“) zu realisieren, weil ein entsprechendes Unterprogramm in üblichen Universalprozessoren nicht schnell genug ist. Einer dieser zentralen Funktionsblöcke ist ein Multiplizierer, mit dem zwei Polynome parallel miteinander multipliziert werden. Die gewünschte Signalauflösung bestimmt dabei die Wortbreite und diese die Ordnung des zugehörigen Galois-Körpers. Ab einem Galois-Körper 7. Ordnung wird der Aufwand für derartige Parallelmultiplizierer sehr groß, denn es sind gleichzeitig 7×7 logische Operationen entsprechend der Wortbreite von 7 Bit durchzuführen. Es handelt sich dabei um einen speziellen Multiplizierer, der in der Art einer Modulo-Operation als resultierende Elemente der Multiplikation immer wieder nur Elemente des endlichen Körpers bildet. Das Array dieser Multiplizierer vereinfacht sich dadurch gegenüber dem Array eines normalen Parallelmultiplizierers. Die logischen Verknüpfungen der einzelnen Partialprodukte bei dieser Modulo-Operation entsprechen der „Exklusiv-Oder“-Verknüpfung (=XOR) und der logischen „Und“-Verknüpfung (=AND). Trotz dieser Vereinfachung stellt der Modulo-Multiplizierer einen erheblichen Aufwand innerhalb des GK-Prozessors dar.

[0004] Es ist daher Aufgabe der Erfindung einen vereinfachten GK-Prozessor anzugeben, mit dem mindestens zwei mit unterschiedlichen Reed-Solomon Codes umgesetzte Datenströme in Echtzeit dekodiert werden können.

[0005] Die Lösung der Aufgabe erfolgt nach dem Merkmalen des Anspruchs 1. Der Grundgedanke der Erfindung besteht darin, für die Reed-Solomon Dekodierung des einen Datenstromes die isomorphen Abbildungseigenschaften von Galois-Körpern gleicher Ordnung auszunutzen. Dies ist dann möglich, wenn die beiden Körper durch unterschiedliche primitive Polynome gleichen Grades definiert sind. Durch diese Transformation ist nur ein einziger Reed-Solomon Dekodierer für beide Datenströme erforderlich, wobei im einzigen GK-Prozessor dann die unterschiedlichen Parameter für die jeweiligen Generator-Polynome von einer Steuereinrichtung eingestellt werden müssen. Selbstverständlich

können auch weitere Datenströme auf gleiche Weise isomorph abgebildet und danach dekodiert werden sofern deren Galois-Körper die gleiche Ordnung wie die beiden ersten Galois-Körper aufweisen.

[0006] Die Erfindung und vorteilhafte Ausgestaltungen werden nun anhand der Figuren der Zeichnung näher erläutert:

Fig. 1 zeigt die Erfindung schematisch als Blockschaltbild mit typischem Umfeld,

Fig. 2 zeigt die Herleitung der Elemente eines ersten primitiven Polynoms,

Fig. 3 zeigt die Herleitung der Elemente eines zweiten primitiven Polynoms und

Fig. 4 zeigt in Tabellenform die Suche nach Lösungen für die isomorphen Abbildung.

[0007] Fig. 1 zeigt schematisch als Blockschaltbild eine Einrichtung 1 für einen Reed-Solomon Dekodierer innerhalb eines GK-Prozessors, der zwei unterschiedliche, Reed-Solomon kodierte Datenströme d1, d2 dekodiert. Die Quelle 2 für die beiden Datenströme d1, d2 ist beispielsweise ein TDM-Signal-Demultiplexer (TDM = Time Division Multiplex), der die beiden Datenströme aus einem Datenstrom dq herausfiltert, der mit einem vorgegebenen Datenformat von einem QPSK-Demodulator 3 (QPSK = Quadrature Phase Shift Keying) geliefert wird. Die Herausfilterung des ersten und zweiten Datenstromes d1, d2 aus dem Datenstrom dq ist in der Anordnung von Fig. 1 mittels eines elektronischen Umschalters 20 angedeutet, dessen Steuereingang vom TDM-Signal-Demultiplexer 2 in Abhängigkeit von der Kopfinformation des Datenstroms dq geschaltet ist. Zudem sucht der TDM-Signal-Demultiplexer 2 aus der kanalweise übertragenen Information im Datenbereich des Datenstroms dq für den ersten Datenstrom d1 nur diejenigen Daten aus, die dem gewünschten Kanal zugeordnet sind. Der QPSK-Demodulator 3 bestimmt, in welchem der vier möglichen Quadranten der Phasenwert des empfangenen Signals dd liegt. Aus der Quadrantenzuordnung werden dann die zugehörigen Symbole bestimmt, deren Daten in Form eines kontinuierlichen Datenstroms dq das digitale TDM-Signal enthalten. Der QPSK-Demodulator 3 und die nachfolgenden Stufen verarbeiten im Ausführungsbeispiel von Fig. 1 digitalisierte Signale. Die Digitalisierung erfolgt in einem Analog-Digital-Umsetzer 4, der von einem analogen Ausgangssignal da eines Tuners 5 gespeist ist und als Ausgangssignal ein digitalisiertes Signal dd liefert, dessen Taktrate mindestens doppelt so hoch wie die Symbolrate ist. Beispielsweise ist für eine Symbolrate von 4 MHz eine Taktrate von mindestens 8 MHz erforderlich und die Auflösung sollte mindestens 32 Stufen, also 5 Bit umfassen.

[0008] Die Schaltungsblöcke des GK-Prozessors enthalten für den ersten und zweiten Datenstrom d1, d2 als wesentliche Einheit einen Reed-Solomon Dekodierer 6. Im Signallauf davor liegt für den ersten Datenstrom d1 eine Zwischenverarbeitungsstufe 7, 8. Für den zweiten Datenstrom d2 liegt vor dem Reed-Solomon Dekodierer 6 eine Zwischenverarbeitungsstufe 9, 10 und ein erster Codeumsetzer 11, der den Datenstrom d3 an seinem Eingang isomorph in einen Datenstrom d4 umsetzt, mit dem dann der Reed-Solomon Dekodierer 6 gespeist ist. Der isomorphen Abbildung vor dem Reed-Solomon Dekodierer 6 entspricht eine inverse, ebenfalls isomorphe Abbildung des Datenstromes d6 in einem Codeumsetzer 12 nach dem Reed-Solomon Dekodierer, mit der die isomorphe Abbildung wieder rückgängig gemacht wird, nachdem die Reed-Solomon Dekodierung stattgefunden hat. Der zugehörige Datenstrom d7 wird in einem statischen Speicher 13 gespeichert. Dieser ist vorzugsweise ein Teil einer Steuereinrichtung 14, die in Abhängigkeit von diesen Daten Steuerbefehle c1, c2 für die Gesamtanordnung erzeugt und auch die jeweiligen Parameter dr Generatorpolynome im Reed-Solomon Dekodierer vorgibt. Die Steuereinrichtung 14 ist vorzugsweise ein integrierter Mikro-Controller (=µC), damit die vom ersten Datenstrom d1 abhängigen Steuerbefehle c1, c2 möglichst ohne Verzögerung zur Verfügung stehen. Dies ist besonders wichtig für die Einstellung der Selektionsparameter im TDM-Signal-Demultiplexer 2, der die relevanten Daten für den ersten und zweiten Datenstrom d1, d2 aus dem wesentlich umfangreicheren QPSK-Demodulator-Ausgangsstrom dq herausfiltern muß. Der Ausgangsdatenstrom d5 des Reed-Solomon Dekodierers 6, der die eigentliche Information, beispielsweise das BC-Signal, enthält, ist einem Signalprozessor 15 zugeführt, der mit weiteren Schaltungsteilen, beispielsweise den Verstärkern und Wiedergabeeinrichtungen, die in Fig. 1 nicht dargestellt sind, verbunden ist.

[0009] Die erste und zweite Zwischenverarbeitungsstufen 7, 8 bzw. 9, 10 können Zeitausgleichstufen oder Puffereinrichtungen 7, 9 für die jeweiligen Datenströme d1, d2 enthalten. Wenn der erste und zweite Datenstrom d1, d2 zusätzlich nach dem Viterbi-Verfahren kodiert sind, dann findet in der ersten und zweiten Zwischenverarbeitungsstufe mittels eines ersten und zweiten Viterbi-Dekodierers 8, 10 eine Viterbi-Dekodierung statt, wobei ebenfalls Schaltungen für den Zeitausgleich oder die Pufferung erforderlich sind. Da Viterbi-Dekodierer relativ umfangreiche Schaltungen enthalten und in jedem Datenblock zunächst der zweite Datenstrom d2 zum Gewinnen der nachfolgenden Steuersignale c1, c2 verarbeitet werden muß, ist es für den erforderlichen Schaltungs- und damit Flächenaufwand bei der monolithischen Integration von Vorteil, für beide Datenströme d1, d2 nur einen einzigen Viterbi-Dekodierer 16 zu verwenden. Um den relativ raschen Datenstrom d2 an einen relativ langsamen Viterbi-Dekodierer 10 bzw. 16 anzupassen, ist in Fig. 1 die Puffereinrichtung 9 vorgesehen, die als FIFO-Speicher (= First-In-First-Out) den langsameren und damit angepaßten Datenstrom d8 bildet.

[0010] In Fig. 2 ist als Erläuterungsbeispiel in Form einer Tabelle die Herleitung der Elemente eines ersten primitiven Polynoms $p_1(x) = x^3 + x + 1$ dritter Ordnung dargestellt. Die unterschiedlichen Elemente des zugehörigen Galois-Körpers

dritter Ordnung werden gebildet, indem das erste primitive Polynom $p_1(x) = 0$ gesetzt wird. Daraus folgt, daß das primitive Element des Galois-Körpers durch die Beziehung $x^3 = x + 1$ definiert ist. Das Einsetzen von "0", "1" oder " α " in diese Gleichung ergibt die einzelnen Elemente, wobei sich diese wegen der Endlichkeit des Galois-Körpers in gleicher Reihenfolge periodisch wiederholen. Da die einzelnen Elemente der Galois-Körper bei der elektronischen Umsetzung üblicherweise als binäre Daten verarbeitet werden, ist eine binäre Schreibweise für die einzelnen Elemente des jeweiligen Galois-Körpers zweckmäßig, wobei sich die binäre Schreibweise an die einzelnen Wertigkeiten von x des primitiven Polynoms anlehnt. Beim Galois-Körper dritter Ordnung lassen sich somit alle Elemente durch eine dreistellige Binärzahl darstellen. Mit diesen Binärzahlen wird auch im Sinne einer EXOR-Logik gerechnet, beispielsweise im Galois-Parallelmultiplizierer. Für den ganzen Galois-Körper steht in Kurzform daher die Schreibweise $GK(2^n)$, wobei mit " n " die Ordnung gemeint ist. Dies hat auch den Vorteil, daß die Anzahl der unterschiedlichen Elemente sofort erkennbar ist. Bei $GK(2^3)$, einem Galois-Körper dritter Ordnung, sind dies ohne das Null-Element "000" sieben unterschiedliche Elemente. Durch die logischen Modulo-Operationen im Reed-Solomon Dekodierer 6 wird dieser Bereich der sieben Elemente nie verlassen.

[0011] Die Tabelle Fig. 2 zeigt in der linken Spalte " $GK_1(2^3)$ " als aufsteigende Potenzen α^i aufeinanderfolgende Elemente des primitiven Polynoms $p_1(x) = x^3 + x + 1$. Die mittlere Spalte " $x^3 = x + 1$ " zeigt die zugehörigen Bildungsgleichungen, die jedoch auf EXOR-Verknüpfungen beruhen. Die rechte Spalte "3 Bit-Form" zeigt die drei Bit des Galois-Elements, das in diesem Format als Datenwort im Reed-Solomon Dekodierer 6 verarbeitet wird. In der mittleren Spalte sind diejenigen Gleichungsteile der leichteren Verständlichkeit wegen unterstrichen, die sich aus der Nullsetzung des primitiven Polynoms mit $p_1(x) = 0$ ergeben, nämlich $x^3 = x + 1$. Obwohl die einzelnen Elemente von Null aufsteigend entsprechend den Potenzen von α größer werden, zeigt die rechte Spalte mit den binär dargestellten Elementen diese stetig aufsteigende Werte im Sinne der üblichen Zweierpotenzahlen nicht. Abweichungen ergeben sich nach "010", "110", "110", "011" und "110". Das ist aber völlig in Ordnung, denn es handelt sich bei diesem dreistelligen Datenformat nicht um Dualzahlen und ihre zugehörigen Wertigkeiten, sondern lediglich um eine datenförmige Zuordnung zu den einzelnen Elementen. Nach dem letzten Element "101" des Galois-Körpers $GK_1(2^3)$ folgt im Sinne der Modulo-Operation als nächstes Element wieder das erste Element "001", wobei das Null-Element "000" ein Sonderelement bildet und immer ausgelassen wird.

[0012] In Fig. 3 sind als "Tabelle 2" die einzelnen Elemente als aufsteigende Potenzen β^i eines zweiten Galois-Körpers dritter Ordnung $GK_2(2^3)$ dargestellt. Das primitive Polynom $p_2(x)$ zur Bildung dieser Elemente ist $p_2(x) = x^3 + x^2 + 1$. Der Aufbau der Tabelle 2 mit den drei Spalten " $GK_2(2^3)$ ", " $x^3 = x^2 + 1$ " und "3 Bit Form" entspricht dem Aufbau der Tabelle 1, so daß sich eine nochmalige Beschreibung erübrigt. Jedem der sieben Elemente ist im binären 3 Bit Format ein eigenes Datenwort zugeordnet. Wie in Tabelle 1 entsprechen diese Datenworte jedoch nicht aufsteigenden Dualzahlen. Durch das andere primitive Polynom $p_2(x)$ ist auch die Reihenfolge der Datenworte gegenüber der von Tabelle 1 geändert.

[0013] Bei einer isomorphen Abbildung werden die Elemente eines endlichen Körpers in die Elemente eines anderen endlichen Körpers gleicher Ordnung umgesetzt. Dies läßt sich für die nach der Erfindung vorgesehenen Codeumsetzer 11, 12 beispielsweise über eine Tabellenzuordnung durchführen, wenn man die Umsetzung kennt. Die Vorgehensweise, wie man die richtige Umsetzung findet, wird anhand eines einfachen Beispiels erläutert, das die vorhandenen Tabellen 1 und 2 verwendet. Bei der isomorphen Abbildung wird zunächst dasjenige Element α^i des einen Galois-Körpers bestimmt, das in das primitive Polynom des anderen Galois-Körpers eingesetzt den Wert Null ergibt und damit definitionsgemäß dort das primitive Element β^1 dieses Galois-Körpers bildet. Potenzen dieses gefundenen primitiven Elementes ergeben dann immer wieder nur Elemente dieses zweiten Galois-Körpers. Diese Erkenntnis gibt einen Weg an, um die einzelnen Zuordnungen für die isomorphe Abbildung eines vorgegebenen Galois-Körpers zu finden. Im erwähnten Beispiel ist das relativ einfach, weil der Galois-Körper nur sieben Elemente aufweist, die leicht in das primitive Polynom des anderen Galois-Körpers eingesetzt werden können. Wenn die Anzahl der Elemente groß ist, also beispielsweise bei einem Galois-Körper siebter Ordnung, dann ist dieses Einsetzen von Hand nicht mehr möglich. Mittels eines geeigneten Programms kann dieses sukzessive Durchmustern auf einfache Weise automatisch über einen Rechner erfolgen. Wenn bei diesem Durchmustern mehrere Elemente gefunden werden, die in das primitive Polynom eingesetzt den Wert Null ergeben, dann kann jeder dieser Lösungen als ein primitives Element für eine zugehörige isomorphe Abbildung dienen. Allerdings sind nicht alle Lösungen im Hinblick auf den jeweiligen Anwendungsfall gleichwertig. Die optimale Lösung kann über eine Simulation der Reed-Solomon Dekodierung herausgefunden werden.

[0014] In Fig. 4 ist als Tabelle 3 ein Beispiel für dieses sukzessive Einsetzen der sieben Elemente γ^1 bis γ^7 des zweiten primitiven Polynoms $p_2(x) = x^3 + x^2 + 1$ in das erste primitive Polynom $p_1(x) = x^3 + x + 1$ dargestellt, um herauszufinden, für welche Elemente das erste primitive Polynom zu Null wird. Die zu prüfenden Elemente y sind dabei nach steigenden Potenzen γ^i zeilenweise geordnet. Die dritte, fünfte und sechste Zeile mit den primitiven Elementen γ^3 bzw. γ^5 bzw. γ^6 ergibt beim Einsetzen in das erste primitive Polynom $p_1(x) = x^3 + x + 1$ jeweils den Wert Null. Somit stellen diese Elemente γ^3 , γ^5 und γ^6 jeweils ein primitives Element dar. Es wird auch hier darauf hingewiesen, daß es sich bei der Berechnung des ersten primitiven Polynoms $p_1(x)$ bei den Vielfachen und Potenzen von y um Multiplika-

tionen und bei dem "+"-Zeichen um XOR-Verknüpfungen handelt.

[0015] Um die Zuordnung zwischen den Elementen des zweiten primitiven Polynoms und des ersten primitiven Polynoms und damit die Abbildung von GK2 nach GK1 und dazu die rückwärtige Abbildung von GK1 nach GK2 zu erhalten, sind entsprechend den erhaltenen Lösungen γ^3 bzw. γ^5 bzw. γ^6 zunächst die zugehörigen Restklassen zu bilden. Mit der Funktion Restklasse wird die Modulo-Division des angegebenen primitiven Polynoms bezeichnet, wobei der Rest aus dieser Division das Ergebnis ist. Entsprechend dem Exponent i der jeweiligen Lösung γ , ist für jede Modulo-Division der zugehörige Exponent des Elements zu multiplizieren. Im vorliegenden Beispiel mit γ^3 ist der Exponent des gefundenen Lösungswertes 3. Da $GK(2^3)$, also ein GK dritter Ordnung über sieben unterschiedliche primitive Elemente verfügt, ist der Wert 7 für die Modulo-Division maßgebend. Die Schreibweise für diese Modulo-Division ist unterschiedlich, "mod (x, y)" oder "x mod y", das bedeutet beispielsweise mit $x=18$ und $y=7$ die Division $18/7 = 2 + 4/7$. Bei diesem Divisionsbeispiel ist 4 der nicht mehr durch 7 teilbare Rest und damit die Lösung der Modulo-Division. In der oben angegebenen Schreibweise sieht das dann wie folgt aus: $18 \bmod 7 = 4$.

[0016] Die sieben Elemente des gesuchten $GK(2^3)$, die bis jetzt noch nicht gefunden sind, sind formal in folgender Tabelle $T(GK(2^3))$ aufgelistet. Dabei steigen die Exponenten i der Potenzen δ^i von $i=0$ bis $i=6$ an. Die Exponenten $i=7$ und $i=0$ liefern wegen der endlichen Anzahl - es sind wie bereits angegeben sieben unterschiedliche Elemente - der Elemente im Galois-Körper dritter Ordnung das selbe Element δ^0 . Mit " δ " wird das primitive Element eines beliebigen Galois-Körpers dritter Ordnung bezeichnet. Die zugehörigen Elemente δ^i ergeben sich dann mit aufsteigendem i von $i=0$ bis $i=6$ als Tabelle $T(\delta^i)$:

$T(\delta^i): (\delta^0, \delta^1, \delta^2, \delta^3, \delta^4, \delta^5, \delta^6)$

[0017] In vereinfachter Schreibform, bei der nur die Exponenten i der Tabelle $T(\delta^i)$ verwendet werden, lassen sich die aufeinanderfolgenden sieben Elemente eines beliebigen Galois-Körpers dritter Ordnung verallgemeinert wie folgt in Tabellenform darstellen:

$T(GK(2^3)): \{0, 1, 2, 3, 4, 5, 6\}$

Aus dieser Tabelle werden nun über die Restklassenbildung mit den gefundenen Lösungen von γ^3 , γ^5 und γ^6 die Transformationen durchgeführt.

[0018] Ein ausführliches Beispiel zeigt die Transformation aller Feldelemente (i) der oben angegebenen Tabelle $T(GK(2^3))$: $\{0, 1, 2, 3, 4, 5, 6\}$ für die Lösung γ^3 :

| | | |
|---|-----------------------------------|-----|
| $i = 0: \bmod(0 \cdot 3, 7) = \bmod(0, 7) = 0$ | im 3-Bit Format (vgl. Tabelle 2): | 001 |
| $i = 1: \bmod(1 \cdot 3, 7) = \bmod(3, 7) = 3$ | dito | 011 |
| $i = 2: \bmod(2 \cdot 3, 7) = \bmod(6, 7) = 6$ | dito | 101 |
| $i = 3: \bmod(3 \cdot 3, 7) = \bmod(9, 7) = 2$ | dito | 100 |
| $i = 4: \bmod(4 \cdot 3, 7) = \bmod(12, 7) = 5$ | dito | 111 |
| $i = 5: \bmod(5 \cdot 3, 7) = \bmod(15, 7) = 1$ | dito | 010 |
| $i = 6: \bmod(6 \cdot 3, 7) = \bmod(18, 7) = 4$ | dito | 110 |
| $i = 7: \bmod(7 \cdot 3, 7) = \bmod(21, 7) = 0$ | (denn $7 \bmod 7 = 0$) dito | 001 |
| $i = 8: \bmod(8 \cdot 3, 7) = \bmod(24, 7) = 3$ | dito | 011 |
| $i = 9: \bmod(9 \cdot 3, 7) = \bmod(27, 7) = 6$ | dito | 101 |

[0019] Ab $i=7$ wiederholen sich die Feld-Elemente. Die zu der Lösung γ^3 gehörige Tabelle der Feld-Elemente, also die zugehörige Transformations-Tabelle, ist $T1(\gamma^3): \{0, 3, 6, 2, 5, 1, 4\}$.

[0020] Für die drei Lösungen γ^3 , γ^5 und γ^6 gibt es damit folgende drei Transformations-Tabellen, die durch die oben beschriebene Modulo-Division auf gleiche Weise gebildet werden:

$T1(\gamma^3): \{0, 3, 6, 2, 5, 1, 4\}$

$T2(\gamma^5): \{0, 5, 3, 1, 6, 4, 2\}$

$T3(\gamma^6): \{0, 6, 5, 4, 3, 2, 1\}$

[0021] Die Rücktransformation der Tabelle T1(γ^3): { 0, 3, 6, 2, 5, 1, 4 } mit γ^3 , γ^5 und γ^6 über die Modulo-Division ergibt die drei folgenden Rücktransmutations-Tabellen:

T4(γ^3): { 0, 2, 4, 6, 1, 3, 5 }

T5(γ^5): { 0, 1, 2, 3, 4, 5, 6 }

T6(γ^6): { 0, 4, 1, 5, 2, 6, 3 }

[0022] Nur die Rücktransmutations-Tabelle T5(γ^5): { 0, 1, 2, 3, 4, 5, 6 }, die mit γ^5 rücktransformiert wurde, zeigt wieder die ursprünglichen Feld-Elemente T(GK(2^3)): { 0, 1, 2, 3, 4, 5, 6 }. Für die Transformation in der einen Richtung gilt die Abbildungsvorschrift γ^3 und für die Rücktransmutation die Abbildungsvorschrift γ^5 . Es ist dabei unerheblich, in welcher Richtung zwischen dem ersten und zweiten Galois-Körper die Hin- und Rücktransmutation erfolgt. Also wenn die Hintransmutation von GK₁(2^3) nach GK₂(2^3) bzw. von GK₂(2^3) nach GK₁(2^3) nach der Abbildungsvorschrift γ^3 erfolgt, dann muß die Rücktransmutation von GK₂(2^3) nach GK₁(2^3) bzw. von GK₁(2^3) nach GK₂(2^3) nach der Abbildungsvorschrift γ^5 erfolgen.

[0023] Auf ähnliche Weise kann dargestellt werden, daß die Rücktransmutation der zweiten Transformations-Tabelle T2(γ^5): { 0, 5, 3, 1, 6, 4, 2 } mit γ^3 , γ^5 und γ^6 als Feld-Elemente folgende Rücktransmutations-Tabellen ergibt:

T7(γ^3): { 0, 1, 2, 3, 4, 5, 6 }

T8(γ^5): { 0, 4, 1, 5, 2, 6, 3 }

T9(γ^6): { 0, 2, 4, 6, 1, 3, 5 }

[0024] Nur die Rücktransmutations-Tabelle T7(γ^3): { 0, 1, 2, 3, 4, 5, 6 }, die mit γ rücktransformiert wurde, zeigt wieder die ursprünglichen Feld-Elemente T(GK(2^3)): { 0, 1, 2, 3, 4, 5, 6 }. Die Hintransmutation erfolgt über die Abbildungsvorschrift γ^5 und die Rücktransmutation über die Abbildungsvorschrift γ^3 .

[0025] Die Rücktransmutation der dritten Transformations-Tabelle T3(γ^6): { 0, 6, 5, 4, 3, 2, 1 } mit γ^3 , γ^5 und γ^6 ergibt als Feld-Elemente folgende Rücktransmutations-Tabellen:

T10(γ^3): { 0, 4, 1, 5, 2, 6, 3 }

T11(γ^6): { 0, 2, 4, 6, 1, 3, 5 }

T12(γ^6): { 0, 1, 2, 3, 4, 5, 6 }

[0026] Nur die Rücktransmutations-Tabelle T12(γ^6): { 0, 1, 2, 3, 4, 5, 6 }, die mit γ^3 rücktransformiert wurde, zeigt wieder die ursprünglichen Feld-Elemente T(GK(2^3)): { 0, 1, 2, 3, 4, 5, 6 }. In diesem Transformationsfall erfolgt die Hin- und Rücktransmutation mit der gleichen Abbildungsvorschrift γ^6 .

[0027] Die Hin- und Rücktransmutation der einzelnen Elemente läßt sich vereinfacht über eine Restklassentabelle Tr darstellen. Mit der Abbildungsvorschrift γ^5 für die Hintransmutation und den drei Feldelementen $i = 0, 1, 2$, wegen der dritten Ordnung der beiden Galois-Körper GK₁(2^3) und GK₂(2^3), ergibt sich folgende Restklassentabelle Tr1:

Tr1: { 1, $1+x+x^2$, $1+x$ } in Polynomform

Tr1: { 001 111 011 } im 3-Bit Format.

Eine andere Schreibweise für das 3-Bit Format von Tr1 ist eine 3 x 3 Matrix:

| | | |
|---|---|---|
| 0 | 0 | 1 |
| 1 | 1 | 1 |
| 0 | 1 | 1 |

[0028] Die Rücktransmutation ergibt auf ähnliche Weise mit der Abbildungsvorschrift γ^3 und den drei Feldelementen $i = 0, 1, 2$ die Restklassentabelle Tr2:

Tr2: { 1, $1+x+x^2$, $1+x$ } in Polynomform

Tr2: { 001 101 110 } im 3-Bit Format.

[0029] Eine andere Schreibweise für das 3-Bit Format von Tr2 ist eine 3 x 3 Matrix:

| | | |
|---|---|---|
| 0 | 0 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

[0030] Das sind gegenüber den 42-Bit der direkten Tabellendarstellung für die 7 primitiven Elemente mit je 3 Bit und den zugehörigen 7 Adressen mit ebenfalls je 3 Bit nur noch 9 Bit, die für eine Transformation zu speichern sind. Diese Matrixdarstellung für die isomorphe Abbildung am Beispiel eines ersten und zweiten Galois-Körpers dritter Ordnung zeigt, daß die Abbildung auf diese Weise besonders einfach in einem GK-Prozessor realisierbar ist und dort wenig Speicher- und Rechenaufwand erfordert. Bei Galois-Körpern höherer Ordnung ist der Einsparungseffekt durch die Matrixdarstellung noch größer.

Patentansprüche

1. Einrichtung (1) für einen Reed-Solomon Dekodierer (6), die von einer Datenquelle (2) im Zeitmultiplexbetrieb mindestens mit einem ersten und zweiten Datenstrom (d1, d2) gespeist ist, deren Daten jeweils einem ersten bzw. zweiten Reed-Solomon-Code entsprechen, wobei ein zugehöriges erstes und zweites primitives Polynom zueinander unterschiedlich, jedoch von gleicher Ordnung sind,

- mit einem ersten Betriebsmodus,
 - in dem der erste Datenstrom (d1) einem Dateneingang des Reed-Solomon-Dekodierers (6), dessen Dekodierungsverfahren durch das erste primitive Polynom und mindestens ein erstes Generatorpolynom definiert ist,
 - direkt oder
 - über eine erste Zwischenverarbeitungsstufe (7, 8) zugeführt wird,
- mit einem zweiten Betriebsmodus,
 - in dem der zweite Datenstrom (d2) einem ersten Codeumsetzer (11)
 - direkt oder
 - über eine zweite Zwischenverarbeitungsstufe (9, 10) zugeführt wird,
 - wobei der erste Codeumsetzer (11) eine isomorphe Abbildung der dem zweiten primitiven Polynom zugeordneten Daten in dem ersten primitiven Polynom zugeordnete Daten durchführt, und
 - der Ausgang des ersten Codeumsetzers (11) den Dateneingang des Reed-Solomon-Dekodierers (6) speist,
 - ein Datenausgang des Reed-Solomon-Dekodierers (6) im ersten bzw. zweiten Betriebsmodus einen fünften (d5) bzw. einen sechsten Datenstrom (d6) liefert, und
 - der sechste Datenstrom (d6) mittels eines zweiten Codeumsetzers (12),
 - der eine zum ersten Codeumsetzer (11) inverse isomorphe Abbildung der dem ersten primitiven Polynom zugeordneten Daten in dem zweiten primitiven Polynom zugeordnete Daten durchführt, in einen siebten Datenstrom (d7) umgesetzt wird.

2. Einrichtung (1) nach Anspruch 1, dadurch gekennzeichnet, daß die erste und zweite Zwischenverarbeitungsstufe (7, 8; 9, 10) einen ersten bzw. zweiten Viterbi-Dekodierer (8, 10) enthält.

EP 1 109 320 A1

3. Einrichtung (1) nach Anspruch 2, gekennzeichnet durch folgende Merkmale:

- für die erste und zweite Zwischenverarbeitungsstufe (7, 8; 8, 10) ist ein einziger Viterbi-Dekodierer (16) vorhanden,
- im ersten Betriebsmodus ist der einzige Viterbi-Dekodierer (16) mit dem ersten Datenstrom (d1) oder einem von ihm abhängigen zehnten Datenstrom (d10) gespeist und sein Ausgang ist direkt mit dem Eingang des Reed-Solomon Dekodierers (6) verbunden und
- im zweiten Betriebsmodus ist der einzige Viterbi-Dekodierer (16) mit einem vom zweiten Datenstrom (d2) abhängigen achten Datenstrom (d8) gespeist und sein Ausgang ist über den ersten Codeumsetzer (11) mit dem Eingang des Reed-Solomon Dekodierers (6) verbunden.

4. Einrichtung (1) nach Anspruch 2 oder 3, dadurch gekennzeichnet, daß in der zweiten Zwischenverarbeitungsstufe (9, 10) vor dem ersten Viterbi-Dekodierer (8) oder dem einzigen Viterbi-Dekodierer (16) eine Puffereinrichtung (9) für den zweiten Datenstrom (d2) vorhanden ist, die den achten Datenstrom (d8) liefert.

5. Einrichtung (1) nach einem der Ansprüche 1 bis 4, dadurch gekennzeichnet, daß zur Datenumsetzung im ersten und zweiten Codeumsetzer (11, 12) jeweils eine gespeicherte Tabelle oder ein Unterprogramm oder eine Kombination aus einer gespeicherten Tabelle und einem Unterprogramm vorhanden ist.

6. Einrichtung (1) nach einem der Ansprüche 1 bis 5, dadurch gekennzeichnet, daß die Daten des siebten Datenstromes (d7), die über eine Steuereinrichtung (14) der Steuerung des ersten und zweiten Betriebsmodus dienen, in der Einrichtung (1) mittels eines statischen Speichers (13) gespeichert sind.

7. Einrichtung (1) nach einem der Ansprüche 1 bis 5, dadurch gekennzeichnet, daß die Daten des fünften Datenstromes (d5) zur weiteren Verarbeitung einem Signalprozessor (15) zugeführt sind.

Fig. 1

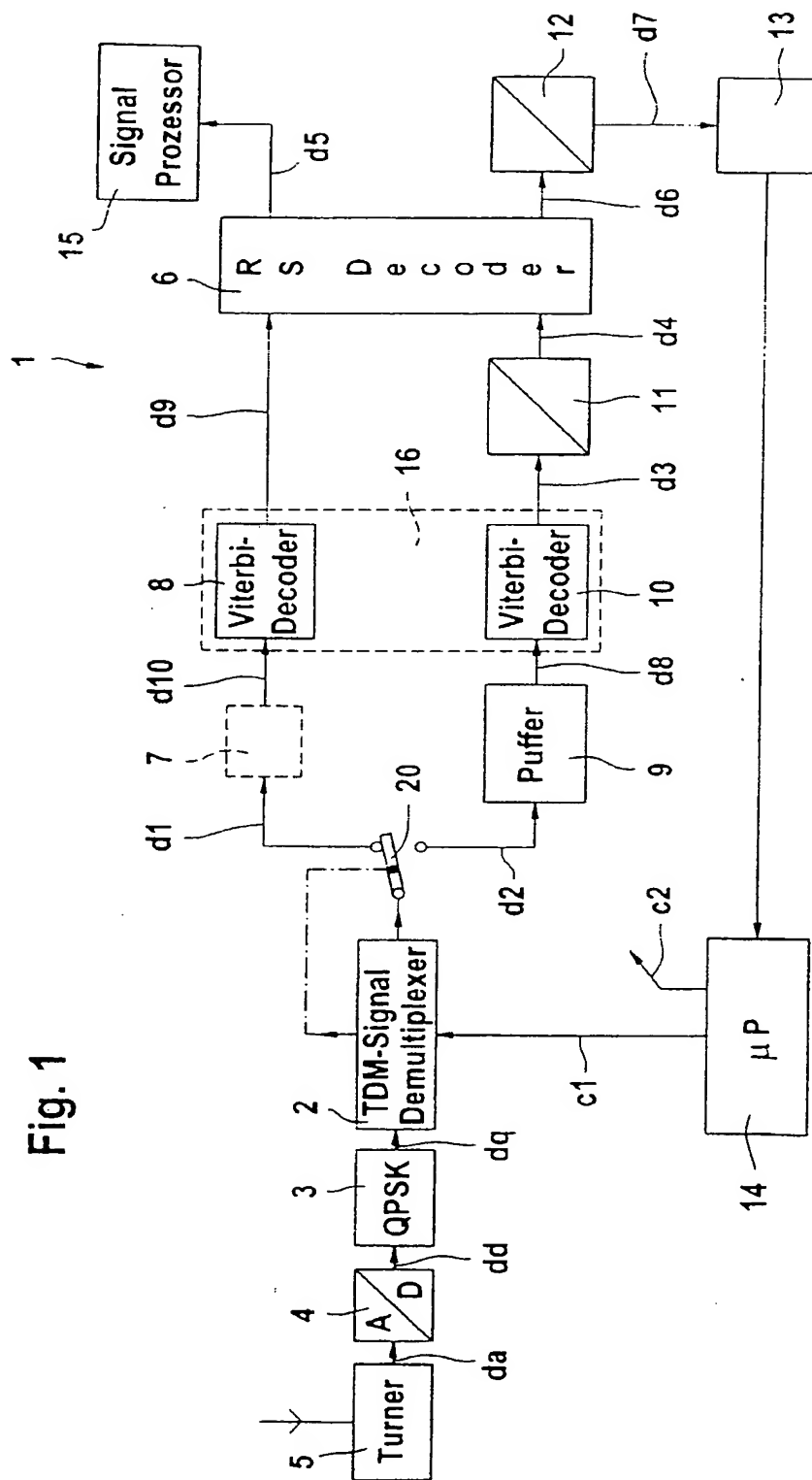


Fig. 2

Tabelle 1

$$p_1(x) = x^3 + x + 1$$

$$p_1(x) = 0$$



$$x^3 = x + 1$$

| <u>GK₁ (2³)</u> | <u>x³ = x + 1</u> | <u>3 Bit-Form</u> |
|---------------------------------------|--|-------------------|
| 0 | 0 | 0 0 0 |
| 1 | 1 | 0 0 1 |
| α | x | 0 1 0 |
| α ² | x · x = x ² | 1 0 0 |
| α ³ | x ³ = <u>x + 1</u> | 0 1 1 |
| α ⁴ | (x + 1) x = x ² + x | 1 1 0 |
| α ⁵ | (x ² + x) · x = x ³ + x ² = <u>x + 1</u> + x ² | 1 1 1 |
| α ⁶ | x ² + x + x ³ = x ² + x + <u>x + 1</u> = x ² + 1 | 1 0 1 |
| α ⁷ | x ³ + x = x + 1 + x = 1 | 0 0 1 |
| α ⁸ | x | 0 1 0 |
| α ⁹ | x ² | 1 0 0 |
| α ¹⁰ | x ³ = <u>x + 1</u> | 0 1 1 |
| . | | |
| . | | |
| . | | |
| . | | |

Fig. 3

Tabelle 2

$$p_2(x) = x^3 + x^2 + 1$$

$$p_2(x) = 0$$



$$x^3 = x^2 + 1$$

| <u>GK₂ (2³)</u> | <u>x³ = x² + 1</u> | <u>3 Bit-Form</u> |
|---------------------------------------|---|-------------------|
| 0 | 0 | 0 0 0 |
| 1 | 1 | 0 0 1 |
| β | x | 0 1 0 |
| β ² | x · x = x ² | 1 0 0 |
| β ³ | x ³ = <u>x² + 1</u> | 1 0 1 |
| β ⁴ | (x ² + 1) · x = x ³ + x = <u>x² + 1</u> + x | 1 1 1 |
| β ⁵ | x ³ + x + x ² = <u>x² + 1</u> + x + x ² = x + 1 | 0 1 1 |
| β ⁶ | (x + 1) · x = x ² + x | 1 1 0 |
| β ⁷ | x ³ + x ² = <u>x² + 1</u> + x ² = 1 | 0 0 1 |
| β ⁸ | x | 0 1 0 |
| β ⁹ | x ² | 1 0 0 |
| β ¹⁰ | x ³ = <u>x² + 1</u> | 1 0 1 |
| . | | |
| . | | |
| . | | |
| . | | |

Fig. 4

Tabelle 3

| $p_1(x) = x^3 + x + 1 = 0$ | | $p_2(x) = x^3 + x^2 + 1 = 0$ | |
|----------------------------|--|---|----------|
| (1) | $\gamma^1: \gamma^3 + \gamma + 1 =$ | $\gamma^2 + 1 + \gamma + 1$ | $\neq 0$ |
| (2) | $\gamma^2: \gamma^6 + \gamma^2 + 1 =$ | $\gamma^2 + \gamma + \gamma^2 + 1$ | $\neq 0$ |
| (3) | $\gamma^3: \gamma^9 + \gamma^3 + 1 =$ | $\gamma^2 + \gamma^2 + 1 + 1$ | $= 0$ |
| (4) | $\gamma^4: \gamma^{12} + \gamma^4 + 1 =$ | $\gamma + 1 + \gamma^2 + \gamma + 1 + 1$ | $\neq 0$ |
| (5) | $\gamma^5: \gamma^{15} + \gamma^5 + 1 =$ | $\gamma + \gamma + 1 + 1$ | $= 0$ |
| (6) | $\gamma^6: \gamma^{18} + \gamma^6 + 1 =$ | $\gamma^2 + \gamma + 1 + \gamma^2 + \gamma + 1 = 0$ | $= 0$ |
| (7) | $\gamma^7: \gamma^{21} + \gamma^7 + 1 =$ | $1 + 1 + 1$ | $\neq 0$ |

$$\boxed{\text{GK}_2(2^3) \xrightarrow[\text{isomorph}]{\gamma^3; \gamma^5; \gamma^6} \text{GK}_1(2^3)}$$



Europäisches
Patentamt

EUROPÄISCHER RECHERCHENBERICHT

Nummer der Anmeldung
EP 00 12 7018

| EINSCHLÄGIGE DOKUMENTE | | | |
|--|--|--|---|
| Kategorie | Kennzeichnung des Dokuments mit Angabe, soweit erforderlich, der maßgeblichen Teile | Betrifft Anspruch | KLASSIFIKATION DER ANMELDUNG (Int.Cl.7) |
| P,Y | EP 1 035 662 A (HITACHI LTD) 13. September 2000 (2000-09-13) * Seite 3, Zeile 20 - Seite 5, Zeile 28; Abbildungen 1-4 * | 1-7 | H03M13/15 H04L1/00 |
| Y | US 5 377 207 A (PERLMAN MARVIN) 27. Dezember 1994 (1994-12-27) * das ganze Dokument * | 1-7 | |
| A | US 4 896 353 A (DEHGANI AYYOUB A ET AL) 23. Januar 1990 (1990-01-23) | | |
| A | US 4 796 110 A (STARK WAYNE ET AL) 3. Januar 1989 (1989-01-03) | | |
| A | US 3 745 528 A (PATEL A) 10. Juli 1973 (1973-07-10) | | |
| | | | RECHERCHIERTE SACHGEBIETE (Int.Cl.7) |
| | | | H03M H04L |
| Der vorliegende Recherchenbericht wurde für alle Patentansprüche erstellt | | | |
| Recherchenamt DEN HAAG | | Abschlußdatum der Recherche 29. März 2001 | Prüfer Devergranne, C |
| KATEGORIE DER GENANNTEN DOKUMENTE | | I : der Erfindung zugrundeliegende Theorien oder Grundsätze E : älteres Patentdokument, das jedoch erst am oder nach dem Anmeldedatum veröffentlicht worden ist D : in der Anmeldung angeführtes Dokument L : aus anderen Gründen angeführtes Dokument & : Mitglied der gleichen Patentfamilie, übereinstimmendes Dokument | |
| X : von besonderer Bedeutung allein betrachtet Y : von besonderer Bedeutung in Verbindung mit einer anderen Veröffentlichung derselben Kategorie A : technologischer Hintergrund O : nichtschriftliche Offenbarung P : Zwischenliteratur | | | |

EPF FORM 1503 03/82 (P04C03)

**ANHANG ZUM EUROPÄISCHEN RECHERCHENBERICHT
 ÜBER DIE EUROPÄISCHE PATENTANMELDUNG NR.**

EP 00 12 7018

In diesem Anhang sind die Mitglieder der Patentfamilien der im obengenannten europäischen Recherchenbericht angeführten Patentdokumente angegeben.
 Die Angaben über die Familienmitglieder entsprechen dem Stand der Daten des Europäischen Patentamts am
 Diese Angaben dienen nur zur Unterrichtung und erfolgen ohne Gewähr.

29-03-2001

| Im Recherchenbericht angeführtes Patentdokument | Datum der Veröffentlichung | Mitglied(er) der Patentfamilie | Datum der Veröffentlichung |
|--|-------------------------------|-----------------------------------|-------------------------------|
| EP 1035662 A | 13-09-2000 | JP 2000252964 A | 14-09-2000 |
| | | CN 1266313 A | 13-09-2000 |
| US 5377207 A | 27-12-1994 | KEINE | |
| US 4896353 A | 23-01-1990 | KEINE | |
| US 4796110 A | 03-01-1989 | AU 6784187 A | 20-08-1987 |
| | | BR 8700714 A | 08-12-1987 |
| | | EP 0234849 A | 02-09-1987 |
| | | JP 62192984 A | 24-08-1987 |
| | | KR 9105557 B | 31-07-1991 |
| US 3745528 A | 10-07-1973 | CA 987031 A | 06-04-1976 |
| | | DE 2263488 A | 05-07-1973 |
| | | FR 2165977 A | 10-08-1973 |
| | | GB 1369725 A | 09-10-1974 |
| | | IT 970967 B | 20-04-1974 |
| | | JP 1071154 C | 30-10-1981 |
| | | JP 48074753 A | 08-10-1973 |
| | | JP 56010657 B | 10-03-1981 |
| | | US RE28923 E | 03-08-1976 |

EPO FORM P4461

Für nähere Einzelheiten zu diesem Anhang : siehe Amtsblatt des Europäischen Patentamts, Nr.12/82